



Missouri University of Science and Technology
Scholars' Mine

Engineering Management and Systems
Engineering Faculty Research & Creative Works

Engineering Management and Systems
Engineering

01 Jun 2008

System of Systems: Power and Paradox

Joseph J. Simpson

Cihan H. Dagli

Missouri University of Science and Technology, dagli@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/engman_syseng_facwork

 Part of the [Operations Research, Systems Engineering and Industrial Engineering Commons](#)

Recommended Citation

J. J. Simpson and C. H. Dagli, "System of Systems: Power and Paradox," *System of Systems Engineering*, Institute of Electrical and Electronics Engineers (IEEE), Jun 2008.

The definitive version is available at <https://doi.org/10.1109/SYSE.2008.4724165>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Engineering Management and Systems Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

System of Systems: Power and Paradox

Joseph J. Simpson

Missouri University of Science and Technology
6400 32nd Ave NW #9
Seattle, WA 98107
jjis-sbw@eskimo.com

Dr. Cihan H. Dagli

Missouri University of Science and Technology
229 Eng. Mgt. & Systems Engineering Dept.
Rolla, MO 65409-0370
dagli@mst.edu

Abstract – Systems concepts and artifacts provide the basis for enumerable sources of power and wealth in our modern world. Culture, art and science all are based on established systems of behavior, values and thought. The current environment is densely populated with physical system artifacts that are used in every aspect of human life. The ubiquitous nature of existing systems has generated a strong interest in using an existing set of systems as the basis for a system of systems. Further interest in the system-of-systems approach is stimulated by rapid development, deployment and expansion of new and existing systems. While successful system and system-of-systems production provides the basis of great power, many system development activities result in failure. The paradox is that while many individual systems work well as a single system, they fail when incorporated as a component of a system of systems. Successful system-of-systems characteristics and attributes are explored in this paper.

Keywords – System, System of Systems, System Success, Paradox, System Failure, System Abstraction Frame, System Abstraction Stack.

1 Introduction

All systems are developed in a dynamic context. Many aspects of the system context are beyond the control and influence of the system owner, architect, engineer and design engineer. The ability of a system, a system component, or a system that is a member of a system of systems to effectively sense and adapt to the dynamic system context is of great value.

Examples of large scale systems that must adapt to dynamic context changes are reviewed in this paper. The main objective of this review is the identification of system attributes and characteristics that facilitate the success of the system component, system and/or system of systems that are constructed and aligned in any given environment.

The dynamic system context and environment will be represented by five general streams of change: the science stream, the technology stream, the application stream, the product stream and the organizational stream. These are

shown in Figure 1, Streams of Change [1]. While this is not an exhaustive list of change types, these five streams of change are deemed sufficient to provide the necessary context for the discussion of the design, development and production of technology-based, engineered systems.

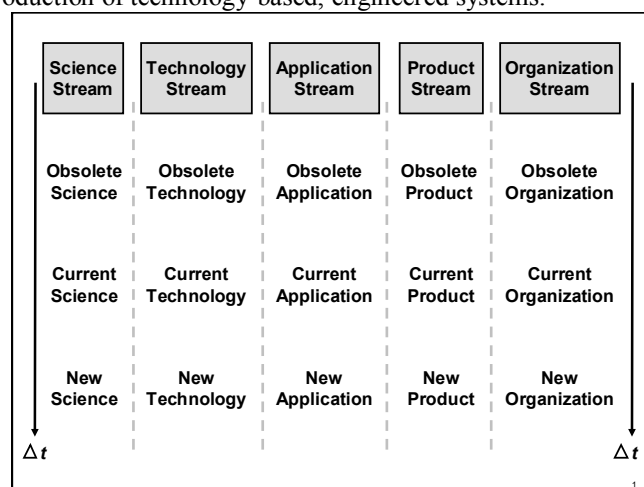


Figure 1 Streams of Change

The position of a system or system of systems can be generally located along the streams of change. This position can be used to set the general system context.

1.1 Abstraction Frames

Using the sequential, flow-of-time context as stated in the introduction, a system evaluation framework has been designed to support the discussion of systems and system-of-systems development. A system can be defined as a relationship mapped over a set of objects [2]. During system design, system architects and engineers use their past experience and combined, technical knowledge to design objects that, when properly integrated into a system or system of systems, produce the customers desired effects. The objects that make up a system must exist before the system can be assembled. The abstraction frame concept was developed to encapsulate the flow of time in one specific context [3]. There can be multiple, concurrent abstraction frames active in any given system development. When objects and/or systems are instantiated in a single abstraction frame, then the time sequence associated with

that abstraction frame applies. Figure 2 shows the sequential flow of time and system development. The system components from the earlier abstraction frames are used in the development of the systems in the later abstraction frames.

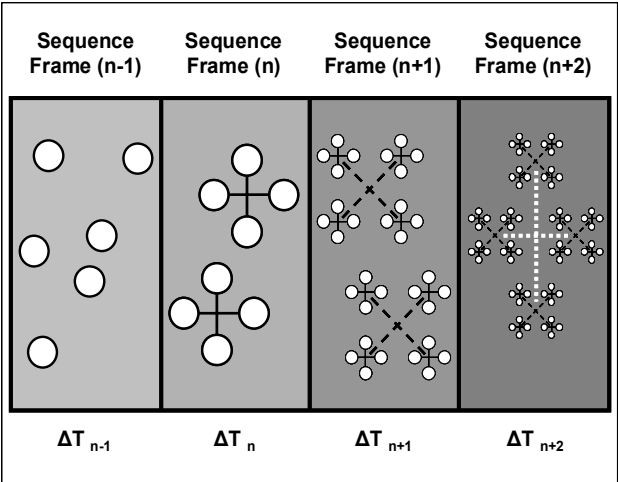


Figure 2 Abstraction Frames

1.2 Abstraction Stacks

A key aspect of system and/or system-of-systems design is the identification and communication of characteristics and attributes at the system-element level. In addition, the system context and the general relationship that relates these system elements between different levels of system abstraction must also be identified and communicated. Abstraction stacks have been developed to facilitate the communication of this type of structured system information [4]. Figure 3 provides a representation of a “house system” that is abstracted along the physical axis of abstraction.

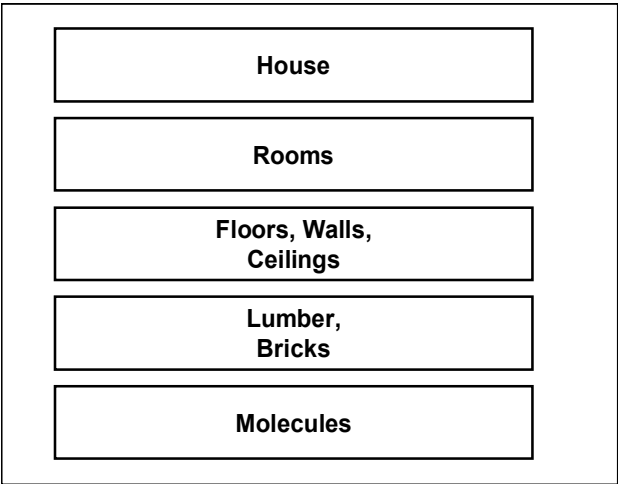


Figure 3 Abstraction Stack

The representation of system behavior and structure is a standard tool in many specialty areas of systems design and integration. There is no common convention or set of abstraction frames that have been established to clearly communicate the semantics of the system design and

development. In this case, it is clear that the axis of system abstraction is the physical axis. The system relationship from the top down can be defined as “is composed of”, while the system relationship from the bottom up is given as “composes.” A fairly small set of standard, system abstraction frames would be able to cover most fundamental driving system relationships and abstraction axes in a comprehensive manner. The development and standardization of a common set of system abstraction frames would greatly enhance the communication of clear, precise system information.

2 System Construction

Given the previously defined concepts in this paper, a system or system of systems must be constructed in a single abstraction frame. The fundamental boundaries of an abstraction frame are time and space, which makes the abstraction frame highly definable. A system production schedule is an example of defining a series of times and contexts that represent specific system abstraction frames. The identification and communication of the controlling system abstraction stacks is much more difficult than the identification and control of system abstraction frames. While a large complex system and/or system-of-systems development will have only one set of abstraction frames (encoded in the program schedule), there may be hundreds of system abstraction frames that are recognized by only specific groups or components of the system development organization.

The classical systems engineering approach controls the system product development using an integrated master schedule as well as strong requirements control and configuration management control. However, in the case of system-of-systems development, this may not be the case. For certain types of system-of-systems development, it may never be possible to apply the level of technical management control that is commonly applied in the classical systems engineering approach.

United States electrical power system deregulation and associated system failures will be used as a first discussion example. The backbone of the United States electrical power grid will be represented by two types of power production facilities: hydroelectric and nuclear power plants. While the main components of the United States hydroelectric system were developed during the 1930’s and 1940’s, the nuclear power plants were mainly developed in the 1960’s to 1980’s. The major programs that developed the hydroelectric production were government programs that used a top down, central control process to design, build and manage these power systems as government controlled utilities. Most of the nuclear power plants were designed and developed by either public or private corporations to produce and sell power to meet customer demand. A large-scale, static power grid was designed to distribute the power to industries and customers in populated areas.

Over a period of time, the power production, distribution, and consumption systems were operationally integrated into a system of systems capable of providing for the power needs of the United States. Then one controlling aspect of this system of systems changed. This change was associated with how power was valued and priced. A power market was developed to buy and sell power to the highest bidder. There was no basic technology change associated with the major system-of-systems change, only a change in the controlling abstraction stack. The original static electrical power distribution system was unable to dynamically adapt to these changes in demand. The electrical power market idea was developed at a high level of abstraction. The ability of the power grid to deliver the produced power to where it was needed was a given part of this high level system abstraction. Therefore, the difference between the age of the basic system components and the controlling system abstraction contributed to system failure. This is an example of a system-of-system failure based on the large time differential between the design and deployment of the basic system building blocks and the controlling system-of-system operational concepts. The controlling system-of-system attributes can be categorized using abstraction frames and abstraction stacks in a manner that highlights system configuration aspects that increase the risk of system failure.

2.1 System Modification

The power distribution system example will now be expanded to facilitate the discussion of another example of system-of-system failure potential. Approximately 50 years after the development of the hydroelectric system in the United States a revolution occurred in computing technology, networking technology and large-scale system monitoring and control. Unlike the centrally planned, top-down government approach that was used to develop the basic components of the power production system, the computer and networking revolution was motivated by private profit, capturing market share, and being a first mover in these developing market areas. These motivations combined with over 20 years of explosive physical infrastructure and system architectural growth has created an integrated set of system of systems that are not naturally secure.

The great value achieved by implementing the initial system control and monitoring mechanisms was not balanced against the future security and liability costs. This cost and benefit system evaluation was not performed for a few basic reasons. One primary reason is the fact that the future state of any system and global impact of any large-scale system is difficult, if not impossible, to predict. Another reason for the absence of a complete cost and benefit analysis lies in the fundamental manner in which the value proposition of a large-scale, networked system is developed over a period of time. When a network is small, it is a lower value target than when the network is large and

well developed. Over the period of time that the large scale system of systems was developed, the environment in which these systems were deployed has changed. These changes include the connection of industrial control nodes to computing systems that are reachable from the open Internet. Another change in the environment is the development of very effective computer node and network node attack procedures and processes. While under initial environmental conditions, the network control system additions were effective and appropriate. As the system environment changed, these network control nodes provided a system attack vector that is open to anyone on the Internet.

2.2 System Disruption and Adaptability

Another area of frequent large-scale, system-of-systems failure is found in the business arena. One specific pattern associated with technology, system and system-of-system development that contributes substantially to business failure has been identified as disruptive technology development [5]. Disruptive technologies start as small scale, niche types of technologies that are too small or insignificant to be addressed by companies that are market leaders in the product and associated technology areas. Companies that are market leaders have value systems and management styles that prevent their interaction with, and adoption of, these disruptive technologies. This fact creates an opportunity for other, more agile and adaptive companies to exploit the disruptive technology to their strong market advantage.

One key example of a disruptive technology is that of open source software products. The extensive availability of open source software products distributed by open source software companies is challenging the market leaders in the closed source software field. Some established companies like the Java Company (Sun Microsystems) and Microsoft have been strongly impacted by open source software. Other companies like IBM have quickly adapted their product lines to take advantage of open source software products, and have seen less of an impact from disruptive open source software technology.

A key observation from the disruptive technology example relates not to the technology or system component type, but directly to the value assigned to the technology by a specific organization. As more organizations acquire and use a given disruptive technology, the value position of the technology becomes greater and applies more pressure on the existing market leaders to engage the disruptive technology. Therefore, the first key attribute required for a successful system-of-systems deployment is adaptability. System flexibility is a related and important system characteristic that is viewed as the ability to adjust to a predictable range of environmental changes. System adaptability is a system characteristic that allows a system to adapt to unknown and unforeseen environmental situations, and is a key primary attribute of successful,

complex system of systems. An adaptable, flexible system of systems must then also have the capability to sense and interact with its environment. This sensing and interactive capability is usually associated with a layered value system and an intelligent control mechanism.

2.3 System Adaptation Interfaces

If uncertain future environmental conditions are a large source of system failure, and adaptation is a key property of a system and/or system of systems that allow the system to effectively cope with an uncertain future, how should a system be designed to take full advantage of the benefits of adaptation? A layered, modular system architecture that exposes multiple system interfaces is proposed as a design mechanism that supports these needed system characteristics and attributes.

A system using the IBM 4758 Common Cryptographic Architecture as a key management component in a hardware security module will be used as an example of an adaptable interface. A security component of Automatic Teller Machines (ATM) was deployed in a hardware configuration. A very large number of the ATM systems in the world use this security component. After the deployment and use of the ATM systems, a small number of successful system security attacks were perfected. The details of these successful attacks spread quickly across the Internet. The changed system environment created a high risk for anyone using this security component to protect and control the distribution of funds.

Addressing this new security threat was constrained by the number of deployed systems and the prohibition against changing any part of the currently deployed physical hardware security system. The key to the successful solution of this problem was found in the interface between the user and the security hardware component. Due to modular system architecture and well-defined, well-documented system interfaces, it was possible to insert another system security component between the user and the legacy security component. In the case referenced here, an artificial neural network approach was proposed to implement an additional security policy layer that addresses the well defined set of successful security attacks [6].

The insertion of the artificial neural network based security policy component at the interface between the user and the static legacy hardware component is an example of adapting a fixed, static system by adding an adaptable, flexible component. This component insertion was directly dependent on existing modular system design and accessible interfaces.

2.4 System Environmental Awareness

The ability to detect, track and understand the changes in the system environment is a valuable characteristic of a system. In a dynamic system of systems, the environmental awareness ability is even more critical because of the

changing nature of the system environment, its components and system-of-systems relationships. A system that is a component of a system of systems controls the connection to the system-of-systems interface. The individual system must determine that being a member of the larger system of systems provides more individual system benefits than the cost of connection and/or membership in the system of systems.

The ability to make and control these types of “connection - no-connection” decisions and actions is a fundamental system-of-system component capability. While an individual system is usually organized around a small set of controlling system relationships combined with a well defined set of abstraction frames, the system relationships and abstraction frame span can vary widely in a system-of-systems configuration. For example, open markets for capital and resources are artificial constructs used by large distributed groups of people to value products, services and resources. A value is also assigned to the liquidity of these items in the market over any given period of time.

As discussed earlier in this paper, a system of systems that is designed based on the capacity of the physical system constrained by laws of nature, may fail dramatically when artificial laws are applied to the evaluation of system connection, control and operation. The fundamental differences between the natural sciences, and sciences of the artificial, creates an area in complex system design, deployment and operation that has a very high potential for generating decisions and actions that directly relate to system-of-system failure.

Natural science is based on the pursuit of knowledge and the discovery, documentation, and communication of the basic unchanging natural laws of the universe. The scientific method has been used to create a large body of scientific knowledge in a top-down fashion. Knowledge and scientific theory were developed and validated at one level of abstraction, without complete understanding, visibility and insight into the layer of knowledge of the next level down in the abstraction stack. Houses were built from wooden timbers before the detailed calculations of force and energy were perfected to enable the practice of modern material science and structural engineering. Structural engineering was developed and applied before the material molecular and atomic forces were completely understood. And science continues to advance as individuals continue to search for, and perfect, the “Theory of Everything” that will describe and communicate the basic unchanging laws of nature [7].

On the other hand, a system of systems is directly dependent on knowledge of the next lower levels of system physical components and/or abstraction stacks to determine when, and the degree to which, the current system-of-system configuration will operate successfully. Further, the span of the required knowledge may be evaluated by first deciding whether the operational science and technology lies in the realm of natural or artificial sciences. This initial

evaluation of the system of systems is used to guide further evaluation and investigation into the system attributes and characteristics that will increase the probability of the development of a successful system-of-systems deployment and operation.

3 Summary

The characteristics and attributes of successful systems and system of systems have been analyzed and evaluated in this paper. System abstraction frames and system abstraction stacks have been presented to facilitate the organization and communication of system context information essential to the successful deployment of system of systems. The following key system attributes and characteristics have been identified as essential components of successful systems: flexibility, adaptability, modular design, open interfaces, and contextual awareness as well as local system control over connection to global system-of-systems resources. Adaptable system interfaces are essential in the construction of system of systems that have the ability to adjust to uncertain future environments.

System interfaces are considered as the primary source of system complexity by several authors. Casti provides an analytical treatment of system complexity, providing a base model of system complexity that assigns the source of complexity to the interface between interacting systems. Casti asserts that the magnitude of the complexity generated by the system interface is a functional value established by one of the systems that participates in the interface [8]. Warfield also assigns the source of complexity to the interface and interaction between systems. Warfield details a scientific approach to system design that is focused on reducing system complexity and large scale system failure [9]. Clearly, a critical aspect in the deployment of successful system of systems is the ability of the system architect and engineers to provide a controllable system design that reduces the complexity associated with the system as its interfaces interact with a wide range of other systems, system components and the system's environment.

4 Conclusions

The power of a large scale system of systems can be realized given the proper system design and operational context. While many of the general characteristics and attributes of successful systems were identified in this paper, further work needs to be accomplished to categorize and evaluate the possible types of system operational environments for any future system. Highlighting the difference between the controlling aspects of natural science and the operational aspects of the artificial sciences in these analysis and evaluation activities will provide further insight into the proper balance for system architectural features.

References

- [1] Joseph J. Simpson, *A Generic, Adaptive Systems Engineering Information Model*, Missouri University of Science and Technology Thesis, Rolla, Missouri, 2004.
- [2] J.J. Simpson and M.J. Simpson, "Systems and Objects," Proceedings CD, Thirteenth Annual International Symposium of the International Council on Systems Engineering, Crystal City, July 2003.
- [3] J.J. Simpson and M.J. Simpson, "System Integration Frameworks," Proceedings CD, Fifteenth Annual International Symposium of the International Council on Systems Engineering, Rochester, July 2005.
- [4] J.J. Simpson and M.J. Simpson, "Formal Systems Concepts," Proceedings CD, Fourth Annual Conference on Systems Engineering Research, Los Angeles, April 2006.
- [5] Clayton M. Christensen, *The Innovator's Dilemma*, Harper Business, Boston, Massachusetts, 1997.
- [6] R.T. Gordon and A. Gehrke, "Controlling Security API Attacks: An Artificial Neural Network Approach," Proceedings of the Artificial Neural Networks in Engineering Conference (ANNIE 2007), St. Louis, November, 2007.
- [7] Herbert A. Simon, *The Sciences of the Artificial*, MIT Press, Cambridge, Massachusetts, 1996.
- [8] John L. Casti, *Alternate Realities, Mathematical Models of Nature and Man*, John Wiley & Sons, New York, New York, 1989.
- [9] John N. Warfield, *A Science of Generic Design: Managing Complexity Through Systems Design*, Iowa State University Press, Ames, Iowa, 1994.